



برنامه‌سازی پیشرفته

تمرین شماره‌ی ۷ - فاز ۳

مدرس: رامتین خسروی

طراحان: امین ستایش، علی زارع، محسن فیاض

مهلت تحویل: جمعه ۱۹ دی، ساعت ۲۳:۵۵

# Smart UTunes.ir!

## مقدمه

شما در فاز های قبلی این پروژه به پیاده‌سازی UTunes پرداختید و به صورت کامل منطق دامنه‌ی آن را پیاده‌سازی کردید. هدف از این فاز پیاده‌سازی واسط کاربری برای فازهای قبلی است. این پیاده‌سازی بر روی بستر وب و مبتنی بر مدل مشتری-سرویس‌دهنده صورت می‌گیرد. برای سهولت، کتابخانه‌ای تحت عنوان APHTTP برای پیاده‌سازی وب سرور در اختیار شما قرار گرفته‌است.



## مدل مشتری-سرویس‌دهنده

مدل مشتری-سرویس‌دهنده ساختار توزیع شده برای طراحی نرم افزار است که در آن نرم افزار به چند بخش مجزا تقسیم می‌شود. یکی از این بخش‌ها سرویس‌دهنده نام دارد که وظیفه‌ی فراهم کردن منابع یا خدمات‌ها را بر عهده می‌گیرد. بخش دیگر که مشتری نام دارد منبع یا سرویس مورد نیاز خود را از سرویس دهنده درخواست می‌کند؛ برای مثال وب سایت eLearn از یک وب سرور تشکیل شده و کاربران با استفاده از مرورگر خود به آن متصل شده و درخواست‌های خود را به آن می‌فرستند. صفحاتی که پس از وارد کردن آدرس <http://elearn.ut.ac.ir> در مرورگر مشاهده می‌شود، جوابی است که از سمت سرویس دهنده در پاسخ به درخواست کاربر فرستاده شده است.

## شرح تمرین

در این فاز شما به پیاده‌سازی یک وب‌سرور بر پایه‌ی پروتکل HTTP برای برنامه‌ی خود که در فازهای قبل پیاده‌سازی کرده‌اید می‌پردازید. مرورگر نیز نقش مشتری را ایفا می‌کند. پروتکل HTTP از تعدادی method پشتیبانی می‌کند که هدف آن‌ها توصیف عملکرد درخواستی از سرور است. دو method ارتباطی مهم در این پروتکل عبارتند از GET و POST که به ترتیب برای درخواست و ارسال اطلاعات استفاده می‌شوند. همچنین پارامترهای پرسمان و بدنه‌ی درخواست برای مشخص کردن اطلاعات ارسال شده از سمت سرویس دهنده مورد استفاده قرار می‌گیرند که هر کدام یک نام و مقدار دارند.

در این فاز لازم است وب‌سرور شما برخی از دستورهایی را که در فازها قبلی پیاده‌سازی کرده‌اید پشتیبانی کند. همچنین باید برای هر یک از این عملیات رابط کاربری مخصوص به آن را با استفاده از زبانِ نشانه‌گذاری HTML پیاده‌سازی کنید.

## آن چه شما باید پیاده‌سازی کنید

در ادامه قابلیت هایی که برنامه‌ی شما باید داشته باشد به تفصیل توضیح داده خواهد شد. شما باید بر اساس هر یک از این قابلیت‌ها یک صفحه برای واسط کاربری آن پیاده‌سازی کنید و با استفاده از کدی که در فازهای قبل زده‌اید، منطق آن را کنترل نمایید.

در هر کدام از صفحه‌ها در صورت رخ دادن استثناهای تعریف شده در فازهای قبل، کاربر به صفحه‌ای با پیام مناسب آن استثنا منتقل شود.

### • صفحه‌ی ثبت‌نام

در این صفحه افراد می‌توانند در سایت ثبت‌نام کنند. این شناسه به طور خودکار در هنگام ثبت‌نام کاربر به او تعلق می‌گیرد.

در صفحه‌ی ثبت‌نام اطلاعات زیر از کاربر گرفته می‌شود:

- نام کاربری
- ایمیل
- رمز عبور

اگر نام کاربری یا ایمیل در سیستم موجود باشد باید متناسب با آن پیغام مناسبی در صفحه‌ی مرورگر به کاربر نشان داده شود.

## • صفحه‌ی ورود

کاربر در این صفحه می‌تواند با وارد کردن نام کاربری و گذرواژه‌ی خود وارد سیستم شود. هر درخواستی که از سمت کلاینت به سرور ارسال می‌شود، با توجه به اینکه کدام کاربر آن را ارسال کرده است، جواب متفاوتی به همراه دارد. برای همین نیاز است تا مشخص شود که درخواست ارسال شده متعلق به چه کاربری است. برای انجام این کار پس از وارد شدن نام کاربری و گذرواژه توسط کاربر و ارسال آن به سرور، در صورتی که این اطلاعات درست باشند، سرور شناسه‌ی یکتای مربوط به کاربر را با عنوان `sessionId` تولید کرده و آن را به کلاینت ارسال می‌کند. از این پس سرور با بررسی `sessionId` هر درخواست و مشخص کردن ایمیل متناظر با آن، متوجه می‌شود که این درخواست متعلق به کدام کاربر است و با توجه به آن به درخواست رسیدگی می‌کند. توجه کنید که کتابخانه‌ی `APTHHTTP` توانایی مدیریت `session Id` ها را دارد. برای اطلاعات بیشتر می‌توانید به بخش **Session**<sup>1</sup> در ویکی کتابخانه مراجعه کنید. بعد از ورود کاربر به صفحه‌ی خانه هدایت می‌شود.

## • خروج

پس از ورود، باید یک دکمه برای خروج در همه‌ی صفحات وجود داشته باشد که به کمک آن، کاربر می‌تواند از سیستم خارج شود. پس از خروج، کاربر باید به صفحه‌ی ورود منتقل شود.

## • صفحه‌ی خانه

در این صفحه لیست آهنگ‌ها شامل شناسه، نام آن، خالق اثر، و سال انتشار آن می‌آید. کاربر با کلیک روی هر کدام از آهنگ‌ها باید به صفحه‌ی مشخصات آن آهنگ هدایت شود.

---

<sup>1</sup> <https://github.com/UTAP/APHHTTP/wiki/Session>



# UTunes

Music For Everyone  
More Than 30 Songs

1. Perfect	Ed Sheeran	2017
2. Morghe Sahar	Mohammadreza Shajarian	1990
3. Shape Of My Heart	Sting	1993
4. Hafteyeh Khaakestari	Farhad	1974

تصویر بالا فقط به عنوان نمونه است و به هر شکل بخواهید می‌توانید این صفحه را طراحی کنید و از طراحی‌های جذاب‌تر استقبال می‌کنیم.  
(ضمناً فقط ۴ آهنگ که در تصویر آمده نیست و تمام آنها باید نمایش داده شوند.)

## ○ اعمال فیلترها (امتیازی)

در صفحه‌ی خانه باید علاوه بر لیستی که توضیح داده شد، موارد لازم برای اعمال سه فیلتر نام خالق اثر، تاریخ انتشار آهنگ و تعداد لایک باشد. برای فیلتر نام خالق اثر، یک جعبه برای گرفتن نام و یک دکمه برای اعمال آن لازم است. برای دو فیلتر دیگر از جعبه‌های ورودی استفاده کنید که فقط عدد می‌پذیرند.

در آخر نیز یک دکمه برای حذف تمام فیلترها باید وجود داشته باشد.  
پس از اعمال هر کدام از فیلترها و یا حذف آنها، باید پیام مناسب را که می‌تواند موفقیت یا عدم موفقیت را نشان دهد به کاربر نمایش دهید.

## • صفحه‌ی مشخصات یک آهنگ

در این صفحه هر کاربر می‌تواند جزئیات یک آهنگ را مشاهده کند.

این مشخصات شامل موارد زیر است:

- نام آهنگ
  - نام هنرمند
  - سال پخش
  - قابلیت پخش آهنگ
  - تعداد کاربرانی که آهنگ را پسندیده‌اند
  - تعداد لیست پخش‌هایی که شامل این آهنگ است
- برای پخش آهنگ با استفاده از لینک داده شده در فایل CSV، می‌توانید از این [لینک](#) کمک بگیرید.
- همچنین در انتهای این صفحه باید ۴ آهنگ جدید براساس پیاده‌سازی انجام‌شده در فاز ۲ به کاربر پیشنهاد شود. در صورتی که تعداد آهنگ‌های قابل پیشنهاد کمتر از ۴ بود، به همان تعداد نشان داده شود.
- این پیشنهادها باید شامل اطلاعات زیر باشد:

- نام آهنگ
  - نام هنرمند
  - سال پخش
- همچنین در این صفحه هر کاربر می‌تواند در صورت تمایل آهنگ را بیسندد و اگر قبلاً آن را پسندیده بود، بتواند آن را از آهنگ‌های پسندیده حذف کند.

## • صفحه‌ی آهنگ‌های پسندیده شده

در این صفحه کاربر می‌تواند لیست آهنگ‌های پسندیده شده خودش را مشاهده کند و با کلیک بر روی هر کدام به صفحه مشخصات آن آهنگ برود.

این لیست باید شامل موارد زیر باشد:

- نام آهنگ
  - نام هنرمند
  - سال پخش
- کاربر باید بتواند در این صفحه آهنگ مورد نظر خودش را از لیست آهنگ‌های پسندیده شده حذف کند.

## • صفحه‌ی مدیریت پلی لیست‌ها

در این صفحه هر کاربر می‌تواند لیست پلی لیست های خودش را مشاهده کند و با کلیک روی هر کدام، به صفحه مدیریت آن پلی لیست برود.

مشخصات زیر باید برای هر پلی لیست نمایش داده شود:

- نام پلی لیست
- وضعیت عمومی یا شخصی بودن

همچنین کاربر در این صفحه باید بتواند یک پلی لیست جدید بسازد. در صورت نیاز می‌توانید یک صفحه جدید برای ساخت پلی لیست ایجاد کنید و کاربر را به آن صفحه منتقل کنید. اطلاعات مورد نیاز برای ساخت یک پلی لیست مانند فاز های قبل نام و وضعیت عمومی/شخصی بودن آن است.

## • صفحه‌ی مدیریت یک پلی لیست

هر کاربر می‌تواند در این صفحه مشخصات یه پلی لیست را مشاهده و در صورت اینکه سازنده پلی لیست بود به آن آهنگ اضافه یا از آن حذف کند.

برای اضافه کردن آهنگ جدید به پلی لیست، کاربر می‌تواند شناسه آهنگ مورد نظر را وارد کند تا به پلی لیست اضافه شود.

همچنین کاربر باید بتواند با کلیک روی هر آهنگ به صفحه مشخصات آن آهنگ برود.

مشخصات پلی لیست شامل موارد زیر است:

- نام پلی لیست
- وضعیت عمومی یا شخصی بودن
- لیست آهنگ‌ها

برای هر آهنگ هم مشخصات زیر باید نشان داده شود:

- نام آهنگ
- نام هنرمند
- سال پخش

## بخش امتیازی

### • CSS<sup>2</sup>

زبان نشانه گذاری CSS یک راه برای تغییر ظاهر صفحه‌هایی است که از HTML بهره می‌برند. استفاده از این زبان علاوه بر این که می‌تواند ظاهر برنامه‌ی شما را بهتر کند برایتان نمره‌ی امتیازی هم به همراه دارد. می‌توانید برای آشنایی بیشتر با CSS به این [لینک](#) مراجعه کنید.

## نحوه‌ی تحویل و نکات

- نحوه اجرا و گرفتن فایل‌های CSV مانند فاز قبل است.

ورودی
<code>./utunes.out &lt;songs_csv_file_relative_address&gt; &lt;liked_songs_csv_file_relative_address&gt;</code>

ورودی نمونه
<code>./utunes.out folder1/folder2/songs.csv folder3/liked_songs.csv</code>

- تمام فایل‌های خود را در قالب یک پرونده‌ی زیپ با نام A7-<SID>.zip در صفحه‌ی Elearn درس بارگذاری کنید که SID شماره‌ی دانشجویی شماست؛ برای مثال اگر شماره‌ی دانشجویی شما ۸۱۰۱۹۸۹۹۹ است، نام پرونده‌ی شما باید A7-810198999.zip باشد.

○ برای مثال، نمونه فایل مورد قبول در زیر آمده است:

A7-810198999.zip

```
|_ ...  
|_ makefile
```

- اطلاعاتی که باید در هر صفحه نشان دهید، همان اطلاعاتی است که در فاز‌های قبل از طریق رابط خط فرمان نشان می‌دادید.

- استفاده از فریمورک‌های دیگر به جز APHTTP مانند NodeJS، Angular، ReactJS و ... مجاز نیست.
- منطق اصلی برنامه شما در فازهای قبل بررسی می‌شود و برای این فاز پیاده‌سازی بخش‌ها گفته شده کافیهست. به جز موارد گفته شده می‌توانید فرض کنید کاربر رفتار معقولی با برنامه دارد.

---

<sup>2</sup> Cascading Style Sheets

- صفحاتی که طراحی می‌کنید باید کاربری‌های گفته شده را داشته باشد ولی طراحی جزئیات هر صفحه به عهده شماست.
- برای مثال برای جابه‌جایی میان صفحات می‌توانید از نوار ناوبری<sup>3</sup> یا هر روش دیگری استفاده کنید.
- بهتر است تا طراحی برنامه‌ی شما طوری باشد که کمترین وابستگی میان منطق برنامه و رابط کاربری آن وجود داشته باشد.
- **دقت کنید** که پرونده زیپ آپلودی شما باید پس از Unzip شدن شامل پرونده‌های پروژه شما (از جمله Makefile) باشد و از زیپ کردن پوشه‌ای که داخل آن فایل‌های پروژه‌تان قرار دارد خودداری فرمایید.
- برنامه‌ی شما باید در سیستم عامل لینوکس و با مترجم g++ با استاندارد c++11 ترجمه و در زمان معقول برای ورودی‌های آزمون اجرا شود.
- **دقت کنید** که پروژه شما باید Multi-file باشد و Makefile داشته باشد. همین‌طور در Makefile خود مشخص کنید که از استاندارد c++11 استفاده می‌کنید.
- درستی برنامه‌ی شما در این فاز توسط دستیاران آموزشی سنجیده می‌شود. زیبایی و راحتی کار با رابط کاربری می‌تواند برای شما نمره‌ی امتیازی داشته باشد.
- **دقت کنید** که نام پرونده‌ی اجرایی شما باید **utunes.out** باشد.
- طراحی درست، رعایت سبک برنامه نویسی درست و تمیز بودن کد برنامه‌ی شما در نمره‌ی تمرین تأثیر زیادی دارد.
- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق قوانین درس با آن برخورد خواهد شد.
- سوالات خود را تا حد ممکن در فروم درس مطرح کنید تا سایر دانشجویان نیز از پاسخ آن‌ها بهره‌مند شوند. در صورتی که قصد مطرح کردن سوال خاص‌تری داشتید، از طریق ایمیل با طراحان این فاز پروژه ارتباط برقرار کنید.

---

<sup>3</sup> Navbar